



Sentiment Analysis

A Probabilistic Approach

S. A. Gieske S. Laan D. S. Ten Velthuis
C. R. Verschoor A. J. Wiggers

Faculty of Science (FNWI)
University of Amsterdam

February 3, 2012

Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion



The goal of the project

Project Description

Performing sentiment analysis on messages about the EO

- Classification Sentiment vs. Non Sentiment
- Classification Positive vs. Negative



Outline

- 1 The goal
- 2 Approach**
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion

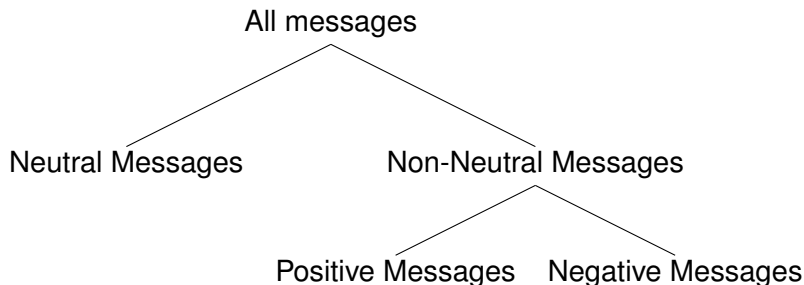


Approach

- Preprocessing of the data
- Perform machine learning algorithms on data
- Use best algorithms to classify real time on server



Hierarchical Classification





Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing**
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion



Outline

3 Data Preprocessing

- Dataset Analysis
- Data Cleaning
- Data Reduction



Dataset Analysis

Dataset messages EO

10.000 messages, 19 features per message

Only 3 features used:

- Source
- Sentiment
- Message contents



Data Cleaning

- Shorten words, e.g. 'saaaaaaai' to 'saaai'
- Stemmer



Data Reduction

- Only use Twitter messages (83% of all messages)
- Remove articles, reference words and prepositions
- Substitute smileys with words
- Remove some punctuation marks (e.g. not ! ?)



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification**
- 5 Webserver Framework
- 6 Conclusion



Outline

4 Classification

- Weighted Sum Probability
- Perceptron
- Support Vector Machine
- Naive Bayes
- Multiclassification with Perceptron
- Entropy
- Neural Network



Classification



Weighted Sum Probability

- Extract features
- Assign sentiment probabilities to features

$$P(feature) = \frac{\sum feature \in C_1}{\sum feature \in C_1 \cup C_2} \quad (1)$$

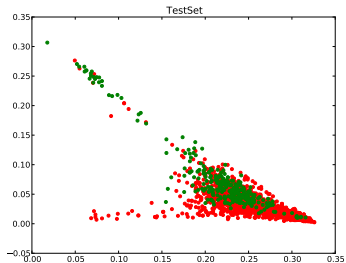
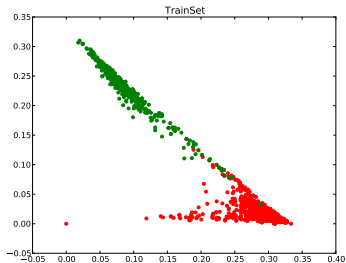
- Assign sentiment probabilities to sentences

$$P(s) = \frac{1}{n} \sum_{f \in s} P(f) \quad (2)$$



Weighted Sum Probability

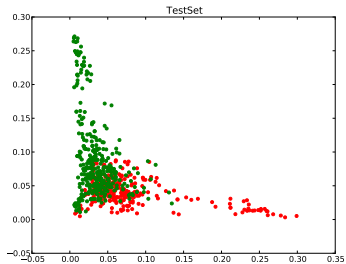
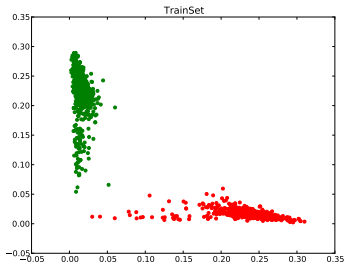
WSP: Neutral vs Non-Neutral





Weighted Sum Probability

WSP: Positive vs Negative





Perceptron

Algorithm

Train linear threshold

Input : Sentence probabilities, sentence values

Output : Treshold



Results & Conclusion

Results : High precision OR recall, never both

Conclusion : Linear threshold not good enough



Support Vector Machine

Algorithm

Fit in featurespace that binds features to classes

Input : Features in vector

Output : Number belonging to class



Results & Conclusion

Results : Not very good recall/precision/accuracy

Conclusion : Fit can not be made on these features
or more data needed to find clear boundary



Naive Bayes

Algorithm

Prior and likelihood lead to posterior

Input : Features from sentence

Output : Probability



Results & Conclusion

		Recall	Accuracy	Precision
Results :	Positive	0.43	0.81	0.39
	Negative	0.40	0.71	0.17
	Neutral	0.61	0.59	0.79

Conclusion : Low recall and precision



Multiclassification with Perceptron

Algorithm

Specialized perceptron for each class, one vs. all

Input : Sentence probability for class

Output : Most likely class



Multiclassification with Perceptron

Results & Conclusion

Results :

Conclusion :



Entropy

Algorithm

Words with highest likelihood for classes

Input : Corpus

Output :



Results & Conclusion

Results :

Conclusion :



Neural Network

Algorithm

Backpropagation

Input : Features from sentence

Output : Value for outputnodes (classes)



Results & Conclusion

Results : Training time = 2.85377444 hours.

Test \ Real	True	False
True	14	6
False	40	94

Conclusion : Still many messages with sentiment incorrectly classified.

Possible cause: ratio of messages with sentiment and nonsentiment.



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework**
- 6 Conclusion



Webserver Framework

Request → Server (PHP/PYTHON) → Result (XML)

Request `http://url.com/?dataset=1&message=De EO is cool!`

Result XML File (Containing: Status, Message, Sentiment, Accuracy, Precision, Recall)



Demo

Action...



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion**



Conclusion

- All learning algorithms have their (dis)advantages
- No satisfying results
- Not enough data

The goal	Approach	Data Preprocessing	Classification	Webserver Framework	Conclusion
		○ ○ ○	○○○ ○○ ○○ ○○ ○○ ○○ ○○		

Questions?